

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-157115

1

(43)Date of publication of application : 31.05.2002

(51)Int.Cl.

G06F 9/42

(21)Application number : 2000-357619

(71)Applicant : HITACHI LTD

(22)Date of filing : 20.11.2000

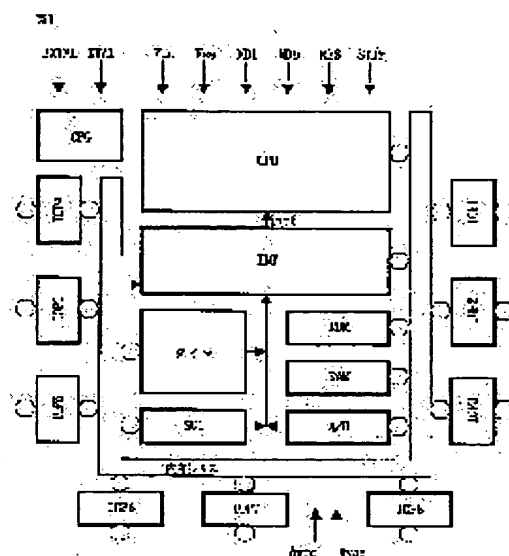
(72)Inventor : MITSUISHI NAOMIKI

(54) DATA PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a means of improving the usability of a data processor such as a microcomputer, reducing program capacity, and cutting down the number of execution states, while minimizing an increase in its logical scale.

SOLUTION: In this data processor, a return command from a sub-routine, or from exception handling, has a command code including information for assigning a plurality of general-purpose registers. When the command code is executed, the assigned registers are returned in addition to the resetting of a program counter. Further, the combination of assigned registers is made to be serial. An evacuation command allowing similar assignment is enabled for the registers.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]



PatentWeb
Home



Edit
Search



Return to
Patent List



Back to
Record



Help

MicroPatent® Worldwide PatSearch: Record 1 of 3

#1

Family of JP2002157115 [How It Works](#)

Family of JP2002157115

No additional family members are found for this document



PatentWeb
Home



Edit
Search



Return to
Patent List



Back to
Record



Help

For further information, please contact:
[Technical Support](#) | [Billing](#) | [Sales](#) | [General Information](#)

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-157115

(P2002-157115A)

(43)公開日 平成14年5月31日(2002.5.31)

(51)Int.Cl.⁷

G 0 6 F 9/42

識別記号

3 3 0

F I

G 0 6 F 9/42

ターミナル(参考)

3 3 0 A 5 B 0 3 3

審査請求 未請求 請求項の数 5 O L (全 19 頁)

(21)出願番号 特願2000-357619(P2000-357619)

(22)出願日 平成12年11月20日(2000.11.20)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 三ツ石 直幹

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体グループ内

(74)代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5B033 AA06 DC08 DD04 DE07

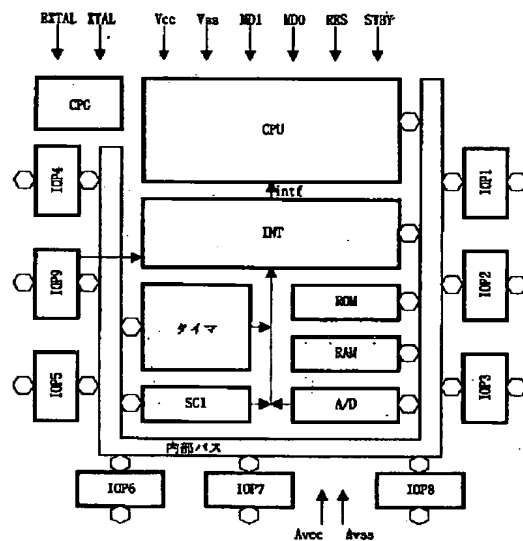
(54)【発明の名称】 データ処理装置

(57)【要約】

【課題】マイクロコンピュータなどのデータ処理装置、その論理的規模の増加を最小限にしつつ、使い勝手を向上し、プログラム容量を縮小したり、実行ステート数を短縮したりする手段を提供すること。

【解決手段】データ処理装置において、サブルーチンから、または例外処理からの復帰命令において、当該命令コードに複数の汎用レジスタを指定する情報を含み、当該命令コードの実行時に、プログラムカウンタの復帰に加えて、前記指定した汎用レジスタの復帰を行うようにする。更に、指定する汎用レジスタの組合せを連続したものにする。同様の指定を可能にした、汎用レジスタの退避命令を実行可能にする。

図1



【特許請求の範囲】

【請求項1】所定の命令を順次実行するデータ処理装置であって、複数のレジスタ手段を有し、スタックからリードした内容に基づく命令のリード動作（PCの復帰）と、命令コード中に指定された情報に従って、スタックからリードした別の内容を、前記レジスタ手段の内、少なくとも1つに書込む動作とを行う復帰命令（RTS/2、RTE/2）の命令実行手段を有することを特徴とするデータ処理装置。

【請求項2】請求項1記載のデータ処理装置であって、レジスタ指定手段と算術演算手段（RINC）を含み、前記復帰命令は、スタックからリードした別の複数の内容を、前記レジスタ手段の内、命令コード中に指定された情報と、算術演算手段とによって指示される複数の書込む動作とを行うことを特徴とするデータ処理装置。

【請求項3】請求項1乃至2記載のデータ処理装置であって、書込むレジスタ手段の数の異なる複数の復帰命令の命令実行手段を有することを特徴とするデータ処理装置。

【請求項4】請求項1乃至3記載のデータ処理装置であって、スタックからリードした内容に基づく命令のリード動作（PCの復帰）を行う別の復帰命令の命令実行手段を有することを特徴とするデータ処理装置。

【請求項5】請求項2乃至4記載のデータ処理装置であって、命令コード中に指定された情報と、算術演算手段とによって指示される複数のレジスタ手段のスタックへのライトを行う転送命令（STM）の命令実行手段を有することを特徴とするデータ処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データ処理装置に関し、特に、半導体集積回路装置によって構成されるマイクロコンピュータにかかり、これらの中央処理装置（CPU）など、及びこれらを用いたデータ処理システム、更に、この開発装置に利用して有効な技術に関するものである。

【0002】

【従来の技術】半導体集積回路装置の製造技術の高度化に伴って、半導体単結晶からなるシングルチップに、中央演算処理装置（Central Processing Unit；以下、単にCPUと称する）、プログラムを格納するROM（Read Only Memory）、書き替え可能に各種データを格納するRAM（Random Access Memory）等を含む構成要素を集積して製造した、マイクロコンピュータが広範囲に普及してきており、種々の目的のデータ処理装置として使用されてきている。このマイクロコンピュータは、CPUが同時に処理し得る情報の量によって性能が異なり、例えば4ビット、8ビット、16ビット、32ビット等のマイクロコンピュータとして区分されてい

る。

【0003】このようなマイクロコンピュータは、アドレス空間の拡張や、命令セットの拡大、高速化等が図られてきている。また、CPUは、ソフトウェアによってその性能が定義されているから、前記のようにアドレス空間の拡張や、命令セットの拡大、高速化等を図ったマイクロコンピュータにおいても、既存のマイクロコンピュータのソフトウェア資産を有効に利用できることが望ましい。

【0004】このため、オブジェクトレベルで互換性を保ちつつ、アドレス空間の拡張や、命令セットの拡大、高速化等を実現した例として、例えば本発明者が先に提案した特開平6-51981号公報がある。これは16ビット単位の変長命令で、いわゆる汎用レジスタ型、或いはロードストア型アーキテクチャを採用することが示されている。

【0005】プログラム用のメモリであるROMを内蔵した、シングルチップ型のものにあつては、内蔵ROMの容量が、外部にメモリを接続するのに比較して、少ないから、プログラム容量を削減することが望ましい。

【0006】16ビット単位の変長命令を採用した場合、オペレーションコードのみを持ち、実効アドレスの指定を行わず、かつ有意義な命令として、復帰（RTS/RTE）命令がある。かかる復帰命令を16ビット単位の命令に割当てると、16ビットの命令コードの全てのビットが利用されていない場合が多い。

【0007】例えば、命令コードH'5470がサブルーチンからの復帰命令であつて、H'5471、H'5472、H'5473が未定義（他の命令として定義されていない）とされているような場合、ビット0、ビット1は利用されていない。

【0008】一方、本発明者らが先に提案した特開平8-263290号公報に記載のCPUは、命令を実行する実行手段を制御する制御手段に指定可能な複数の汎用レジスタの組合せを固定にし、複数の汎用レジスタのスタックに対する退避／復帰命令を有し、複数の汎用レジスタを順次待避／復帰するようにしている。汎用レジスタの組合せを固定にすることによって、汎用レジスタの指定に要するビット数を節約できる。

【0009】かかる退避／復帰命令は、サブルーチンや例外処理などの、処理の切れ目で、それ以前の処理の状態を保存する目的に用いられ、アドレッシングモードは、スタックポインタのアリデクリメント（待避）／ポストインクリメント（復帰）にが使用できる。

【0010】つまり、かかる汎用レジスタの復帰命令は、前記サブルーチンまたは例外処理からの復帰命令（RTS/RTE）の前に置かれることが多い。

【0011】いわゆるアキュムレータ型アーキテクチャのCPUにおいては、サブルーチンへの分岐／サブルーチンからの復帰時に、PCに加えて、アキュムレータの

退避／復帰を行う場合がある。これは、データの処理が少数のアクムレータ上で行われ、サブルーチンのように処理が変わるときには、アクムレータを空ける必要があるためである。換言すれば、アクムレータを自動的に退避／復帰を行っても、無駄にならないからである。

【0012】これに対して、汎用レジスタ型アーキテクチャのCPUにおいては、汎用レジスタのいずれにデータを置くかについて、自由度が高く、また、サブルーチンへの分岐／サブルーチンからの復帰時の引数が、汎用レジスタに割当てられる場合もある。このため、所定の（全部または一部の）汎用レジスタの退避／復帰を自動的に行うと、無駄になりやすい。不所望の汎用レジスタの退避／復帰によって、実行ステート数や、スタックの使用量が増加してしまうからである。

【0013】マイクロコンピュータのプログラムはC言語で記述されることが多くなっている。C言語で多く用いられる関数は、サブルーチンとして実現される。

【0014】

【発明が解決しようとする課題】本発明が解決しようとする問題点は、マイクロコンピュータなどのデータ処理装置、その論理的規模の増加を最小限にしつつ、使い勝手を向上し、プログラム容量を縮小したり、実行ステート数を短縮したりする手段を提供することにある。

【0015】特に、C言語などを使用したプログラムにおいて、関数を多用した場合のプログラム容量の縮小、実行ステート数の短縮を図ることである。

【0016】また、既存のソフトウェア資産を有効に利用可能にし、使用者の開発効率を向上するとともに、開発装置を共通に利用可能にし、開発装置の開発効率を向上するとともに、いち速く開発環境を提供可能にするこ

【0017】本発明の前記ならびにそのほかの目的と新規な特長は、本発明書の記述及び添付図面から明らかになるであろう。

【0018】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0019】本発明のデータ処理装置は、サブルーチンから、または例外処理からの復帰命令において、当該命令コードに複数の汎用レジスタを指定する情報を含み、当該命令コードの実行時に、プログラムカウンタの復帰に加えて、前記指定した汎用レジスタの復帰を行うようにする。

【0020】更に、指定する汎用レジスタの組合せを連続したものにする。

【0021】同様の指定を可能にした、汎用レジスタの退避命令を実行可能にする。

【0022】上記した手段によれば、サブルーチンか

ら、または例外処理からの復帰命令と、汎用レジスタの復帰命令を同一の命令（例えば16ビット）とすることにより、使い勝手を向上し、プログラム容量を短縮できる。

【0023】16ビット単位の可変長の命令コード体系を採る場合にも、命令プリフェッチや命令デコードを簡易にしつつ、命令コードを有効に利用して、全体的なプログラム効率を向上できる。

【0024】同様の指定を可能にした、汎用レジスタの退避命令を持ち、これをサブルーチン乃至例外処理ルーチンの先頭で実行し、最後に、本発明の復帰命令を実行することによって、より一層の使い勝手を向上できる。

【0025】指定する汎用レジスタの組合せを固定にすることによって、汎用レジスタを指定する情報を短くでき、限られた命令コード（例えば16ビット）であっても、指定可能にできる。

【0026】汎用レジスタの復帰と、プログラムカウンタなどの復帰を連続的に行うことによって、内部の動作を容易にし、高速化できる。

【0027】

【発明の実施の形態】図1に、本発明の適用されたマイクロコンピュータのブロック図を示す。

【0028】マイクロコンピュータは、シングルチップ型であり、全体の制御を司るCPU、割込コントローラ（INT）、CPUの処理プログラムなどを格納するメモリであるROM、CPUの作業領域ならびにデータの一時記憶用及びスタック用のメモリとすることができるRAM、タイマ、シリアルコミュニケーションインタフェース（SCI）、A/D変換器、第1乃至第9入出力ポート（IOP1～IOP9）、クロック発振器（CPG）の機能ブロック乃至はモジュールから構成され、公知の半導体製造技術により1つの半導体基板上に形成される。

【0029】かかるマイクロコンピュータは、電源端子として、グラウンドレベル（Vss）、電源電圧レベル（Vcc）、アナロググラウンドレベル（AVss）、アナログ電源電圧レベル（AVcc）、アナログ基準電圧（Vref）、の他専用制御端子として、リセット（RES）、スタンバイ（STBY）、モード制御（MD0、MD1）、クロック入力（EXTAL、XTAL）端子を有する。

【0030】CPGの端子EXTAL、XTALに接続される水晶発振子またはEXTAL端子に入力される外部クロックに基づいて生成される基準クロック（システムクロック）に同期して、マイクロコンピュータは動作を行う。この基準クロック1周期をステートと呼ぶ。

【0031】マイクロコンピュータの機能ブロックは、内部バスによって相互に接続される。バスの制御を行う、図示はされないバスコントローラを内蔵している。内部バスはアドレスバス・データバスのほか、リード信号・

ライト信号・バスサイズ信号を含み、これらはバスコマンドとしてコード化される。

【0032】かかる機能ブロックやモジュールは内部バスを介して、CPUによってリード／ライトされる。内部バスのデータバス幅は32ビットとする。内蔵ROM、RAMリード／ライトを1ステートでリード／ライト可能とする。

【0033】タイマ、SCI、A/D変換器、IOP1～IOP9、SYSC、INT、CPGなどが有する制御レジスタを総称して、内部I/Oレジスタと呼ぶ。

【0034】各入出力ポートは、アドレスバス、データバス、バス制御信号或いはタイマ、SCI、A/D変換器の入出力端子と兼用されている。即ち、タイマ、SCI、A/D変換器は、それぞれ入出力信号を有し、入出力ポートと兼用にされた端子介して、外部と入出力されるものである。例えばIOP5、6は、タイマの入出力端子と兼用、IOP7はSCIの入出力端子と兼用にされている。アナログデータの入出力(AIN0～AIN7)端子は、IOP8と兼用にされている。

【0035】外部割込み入力、IOP9と兼用にされている。

【0036】かかるマイクロコンピュータにリセット信号RESが与えられると、CPUを始めとし、マイクロコンピュータはリセット状態になる。このリセットが解除されると、モード制御で指定された動作モード(シングルチップ、外部バス拡張、内蔵ROM有効など)とされ、CPUは所定のアドレス(ベクタアドレス)からスタートアドレスをリードして、このスタートアドレスから命令のリードを開始するリセット例外処理を行う。この後、CPUは逐次、ROMなどから命令をリードし、解説して、その解説内容に基づいて、フラグやビットの判定や操作を含む、データの処理或いはRAM、タイマ、SCI、A/D変換器、入出力ポートなどとのデータ転送を行う。即ち、CPUは、タイマ、SCI、A/D変換器、入出力ポートなどから入力されるデータ或いは機器の状態や指示(スイッチ、ボリュームなど)を参照しつつ、ROMなどに記憶されている命令に基づいて処理を行い、その結果に基づいて、入出力ポート、タイマなどを使用しつつ、外部に信号を出力し、機器の制御を行うものである。

【0037】割込みコントローラ(INT)は、タイマ、SCI、A/D変換器、外部入力の各割込み信号を入力して、割込み要求信号をCPUに出力する。

【0038】CPUに割込み要求信号が与えられると、CPUは実行中の処理を中断して、割込み例外処理状態を経て、所定の処理ルーチンに分岐し、所望の処理を行い、割込要因をクリアしたりする。所定の処理ルーチンの最後には、通常復帰命令がおかれ、この命令を実行することによって前記中断した処理を再開する。

【0039】割込み例外処理の実行時には、後述のプロ

グラムカウンタ(PC)やコンディションコードレジスタ(CCR)などがRAMなどに退避され、復帰命令実行時に上記プログラムカウンタ(PC)やコンディションコードレジスタ(CCR)などが復帰される。

【0040】更に、割込み処理ルーチンでは所要の処理を行うため、汎用レジスタを使用する必要がある。割込み発生直前の状態を保存するため、割込み処理ルーチンの先頭で、所要の汎用レジスタをスタックに退避する。これは、例えば、

STM ER0-ER3, @-SP

と記述される。割込み処理ルーチンの最後に、退避した汎用レジスタを復帰してから、復帰命令(RTE)を実行して、割込み発生直前のプログラムを継続する。これは、従来技術では、例えば、

LDM @SP+, ER0-ER3

RTE

と記述される。

【0041】本発明によれば、上記LDMとRTE命令を1命令で実行して、

RTE/4 ER0-ER3

と記述すればよいようにする。

【0042】同様に、サブルーチンにおいても同様に、所要の処理を行うため、汎用レジスタを使用する。例えば、サブルーチンの先頭で、

STM ER0-ER1, @-SP

と記述し、最後に、

LDM @SP+, ER0-ER1

RTS

と記述される。

【0043】本発明によれば、上記LDMとRTE命令を1命令で実行して、

RTS/2 ER0-ER1

と記述すればよいようにする。

【0044】図2に、CPUに内蔵されている汎用レジスタ及び制御レジスタの構成例(プログラミングモデル)を示す。

【0045】CPUは、32ビット長の汎用レジスタを8本持っている。汎用レジスタは、すべて同機能を持っており、アドレスレジスタとしてもデータレジスタとしても使用することができる。

【0046】データレジスタとしては32ビット、16ビット及び8ビットレジスタとして使用可能。アドレスレジスタ及び32ビットレジスタとしては、一括して汎用レジスタER(ER0～ER7)として使用する。16ビットレジスタとしては、汎用レジスタERを分割して汎用レジスタE(E0～E7)、汎用レジスタR(R0～R7)として使用する。これらは同等の機能を持っており、16ビットレジスタを最大16本まで使用することができる。なお、汎用レジスタE(E0～E7)を、特に拡張レジスタと呼ぶ場合がある。8ビットレジスタ

としては、汎用レジスタRを分割して汎用レジスタRH (R0H~R7H)、汎用レジスタRL (R0L~R7L)として使用する。これらは同等の機能を持っており、8ビットレジスタを最大16本まで使用することができる。各レジスタ独立に使用方法を選択することができる。

【0047】汎用レジスタER7には、汎用レジスタとしての機能に加えて、スタックポインタ (SP)としての機能が割当てられており、例外処理やサブルーチン分岐などで暗黙的に使用される。例外処理は前記割込み処理を含む。

【0048】PCは24ビットのカウンタで、CPUが次に実行する命令のアドレスを示す。特に制限されないもののCPUの命令は、すべて2バイト (ワード)を単位としているため、最下位ビットは無効であり、命令リード時には最下位ビットは0とみなされる。

【0049】CCRは8ビットのレジスタで、割込みマスクビット (I)とハーフキャリ (H)、ネガティブ (N)、ゼロ (Z)、オーバフロー (V)、キャリ (C)の各フラグを含む。ハーフキャリ (H)、ネガティブ (N)、ゼロ (Z)、オーバフロー (V)、キャリ (C)の各フラグは転送命令や演算命令実行時に、データを検査して、その状態を反映する。ハーフキャリ (H)は10進補正用にのみ用いられる。

【0050】図3に、CPUの構成例を示す。

【0051】CPUは、制御部と、前記汎用レジスタER0~ER31、プログラムカウンタPC、コンディションコードレジスタCCRを含む実行部から構成される。

【0052】制御部は、命令レジスタIR1、命令デコーダDEC、レジスタセクタRDEC、割込受け付け回路INTを含む。命令デコーダDECは、例えば、マイクロROM或PLA (Programmable Logic Array)または布線論理で構成される。命令デコーダDECの出力の一部が命令デコーダDECにフィードバックされている。これは各命令コード内の遷移に用いるステージコード (TMG)と、命令コード間に用いる制御信号 (MOD)を含む。即ち、前記STM命令の第1ワードは汎用レジスタの本数を指定する制御信号を発生し、割込みマスク信号を出力して、割込みの受付を禁止する。

【0053】割込み制御部INTは、後述の図1の割込みコントローラの出力する割込み要求信号を受け付ける。また、命令デコーダの出力する割込みマスク信号を参照して、割込みがマスクされていなければ、命令変更部CHGに割込みを指示し、割込み例外処理用の命令コードの実行を行わせる。

【0054】実行部には、更に、テンポラリレジスタTR、算術論理演算器ALU、インクリメンタINC、リードデータバッファDBR、ライトデータバッファDB

W、アドレスバッファABを含む。これらのブロックはGBバス、DBバス、WBバスによって相互に接続されている。

【0055】算術論理演算器ALUは、命令によって指定される各種の演算や実効アドレスの計算など用いる。

【0056】インクリメンタINCは、主にPCの加算やアドレス計算に用いられる。

【0057】DBRは、ROM、RAM、内部I/Oレジスタ、或いは図示はされない外部メモリから、リードした命令コードやデータを一時的に格納する。DBWはROM、RAM、内部I/Oレジスタ、或いは外部メモリへのライトデータを一時的に格納する。DBR、DBWによってCPU内部動作と、CPU外部のリード/ライト動作のタイミングを調整している。

【0058】アドレスバッファABは、CPUがリード/ライトするアドレスを一時的に格納する。

【0059】図4に、レジスタセクタRDECの一部の詳細なブロック図を示す。

【0060】レジスタデコーダRDECは、レジスタフィールドを保持するIR1から出力されるレジスタフィールド情報と、命令デコーダDECの制御信号出力を入力して、各汎用レジスタに対応するレジスタ選択信号を発生する。

【0061】レジスタ指定フィールドは2つある (r1、r2)。

【0062】第2のレジスタ指定フィールド (r2)は、IR1のビット3~0と、命令デコーダの制御信号出力 (inc、dec)から生成される。

【0063】上記を入力したインクリメンタ (RINC)の出力が、システムクロックの反転φ#でラッチされ、第2のレジスタ指定フィールド (r2)となる。

【0064】例えば、

STM ER0-ER3, @-SP

の場合、r2の初期値として、0が与えられる。その後、後述の所定のタイミングで、インクリメント信号 (inc)によって、r2は0→1→2→3とされる。

【0065】また、

LDM @SP+, ER0-ER1

または、

RTS/2 ER0-ER1

の場合、r2の初期値として、1が与えられる。その後、後述の所定のタイミングで、インクリメント信号 (dec=r d—)によって、r2は1→0とされる。

【0066】これによって、レジスタ選択回路自体をそのほかの命令と共通化することが、論理的規模の増加を抑止できる。

【0067】r2のビット3は、これらの命令では使用しないため、インクリメンタ (RINC)は3ビットでよい。インクリメンタ (RINC)を持つことにより、

さほど論理規模を大きくすることなく、連続した任意の汎用レジスタの組合せを指定可能にできる。

【0068】なお、その他のRDECの詳細については省略する。

【0069】図5に、複数汎用レジスタの復帰を含む第1の復帰命令(RTS/n)、第2の復帰命令(RTE/n)の命令フォーマットを示す。

【0070】第1の復帰命令(RTS/n)は、サブルーチンからの復帰命令であり、第1の復帰命令(RTE/n)は、例外処理ルーチンからの復帰命令である。

【0071】ビット15～7はオペレーションフィールドであり、復帰命令を指定する。

【0072】ビット6が0のとき、ビット5、4によって、汎用レジスタの本数を指定する。01のときは2本、10のときは3本、11のときは4本とする。汎用レジスタの選択は、連続した2、3、4本の選択が可能である。例えば、4本の場合には、ER0-3、ER1-4、ER2-5、ER3-6、が指定可能である。ER7は、スタックポインタと兼用であり、対象としない。

【0073】ビット6が1のときは、汎用レジスタの復帰を伴わない。即ち、従来技術の復帰命令と同一であり、共通の命令コード(H'5470)とする。

【0074】ビット3～0で最初に使用する汎用レジスタを指定する。ビット3は予約されているものとし、0とする。

【0075】例えば、RTS/4 ER0-3のときは、ER3から復帰されるので、H'5433の命令コードとなる。

【0076】第2の復帰命令(RTE/n)も、オペレーションフィールドが異なるのみであり、前記同様である。

【0077】STM命令の命令フォーマットは、2ワード命令であり、第2ワードのビット3～0で最初に使用する汎用レジスタを指定する。これは上記復帰命令と共通化されている。

【0078】LDM命令の命令フォーマットも同様である。

【0079】図6に、複数汎用レジスタの復帰を含む第1の復帰命令(RTS/n)、第2の復帰命令(RTE/n)のスタックの構成の例を示す。

【0080】スタックはスタックポインタ(SP:ER7)によって、スタックされた最後のアドレスが指示されている。

【0081】サブルーチン分岐命令(例えば、BSR命令)によって、プログラムカウンタがスタックに退避される。その後、サブルーチンの先頭で、所要の汎用レジスタが退避される(例えば、STM ER0-ER1, @-SP)。

【0082】これに対応して、第1の復帰命令(例え

ば、RTS/2 ER0-ER1)は、まず、指定した汎用レジスタの復帰として、スタックからのリードを行い、指定した汎用レジスタ(ER1、ER0)に順次格納する。引続き、プログラムカウンタの復帰として、スタックからのリードを行い、プログラムカウンタに格納し、この内容に従って、命令のリードを行う。

【0083】同様に、例外処理によって、プログラムカウンタとコンディションコードレジスタがスタックに退避される。プログラムカウンタが24ビットであり、コンディションコードレジスタと組合せて、スタックに退避される。その後、サブルーチンの先頭で、所要の汎用レジスタが退避される(例えば、STM ER0-ER3, @-SP)。

【0084】これに対応して、第2の復帰命令(例えば、RTE/4 ER0-ER3)は、まず、指定した汎用レジスタの復帰として、スタックからのリードを行い、指定した汎用レジスタ(ER3、ER2、ER1、ER0)に順次格納する。引続き、プログラムカウンタとコンディションコードレジスタの復帰として、スタックからのリードを行い、プログラムカウンタとコンディションコードレジスタに所定の内容を格納するとともに、命令のリードを行う。

【0085】図7に、第1の復帰命令(RTS/n)の動作フローを示す。

【0086】ステップ1で、制御信号Aで、スタックポインタ(SP)の内容がアドレスバス(IAB)に出力され、ロングワードのリードが指示される。また、SPの内容のインクリメント(+4)が指示される。制御信号Cで、リードしたデータのDBRへの格納が指示される。

【0087】ステップ2～5で、制御信号Bで、前のステップでリードした内容の汎用レジスタへの書込みが指示される。具体的には、一旦、内部バスGBに出力し、ALUを経由して、内部バスWBに出力し、指定された汎用レジスタに書込まれる。レジスタデコードRDECに対してレジスタフィールドのデクリメント(-1)が指示される。前記同様に、制御信号Aで、スタックポインタ(SP)の内容のアドレスバス(IAB)への出力と、ロングワードのリードが指示され、SPの内容のインクリメント(+4)が指示される。制御信号Cで、リードしたデータのDBRへの格納が指示される。ステップ2～4は、指定した汎用レジスタの本数(n)に従って、所定回数を実行される。

【0088】ステップ6で、制御信号Bで、前のステップでリードした内容のプログラムカウンタ(PC)への書込みが指示される。

【0089】ステップ7で、制御信号Aで、命令リードとして、プログラムカウンタ(PC)の内容がアドレスバス(IAB)に出力され、メモリワードリードが指示される。また、PCの内容のインクリメント(+2)が

指示される。制御信号Cで、リードされた命令をIRに格納する。

【0090】ステップ8で、同様に、制御信号Aで、プログラムカウンタ(PC)の内容のアドレスバス(IA B)への出力と、メモリワードリードが指示され、PCの内容のインクリメント(+2)が指示される。制御信号Bで、前のステップでリードした命令(IR)の命令デコーダへの入力指示される。次の命令の実行が開始される。

【0091】PCの復帰時、スタックからリードした内容を、一旦、PCに格納するものとしたが、これを行わ

ず、ステップ7で、PCの代わりに、DBRから読み出すようにしてもよい。

【0092】汎用レジスタの復帰を行わない復帰命令の場合は、ステップ2～5をスキップすればよい。換言すれば、従来技術の復帰命令に、ステップ2～5を追加すればよい。

【0093】かかる動作フローに従って、ハードウェア記述言語(HDL)を記述し、命令デコーダDECを生成することができる。第1の復帰命令(RTS/n)に関する部分は、以下の通りである。

【0094】

```

case x(opcode)                                (1)
16' b01010100_0001????: begin                (2)
    case(tmg)                                  (3)
    4' b0001: begin                            (4)
        nexttmg = 4' b0100;                    (5)
        bcmd = long_read;                      (6)
        ...
    end                                          (7)
    4' b0100: begin                            (8)
        nexttmg = 4' b0101;                    (9)
        bcmd = long_read;                      (10)
        ...
    end                                          (11)
    ...
end                                              (12)
16' b01010100_0010????: begin                (13)
    case(tmg)                                  (14)
    4' b0001: begin                            (15)
        nexttmg = 4' b0010;                    (16)
        ...
    4' b0010: begin                            (17)
        nexttmg = 4' b0100;                    (18)
        ...
    end
end

```

本記述は、IEEE1364として標準化されているHDL記述言語に従っている。かかるHDL記述言語自体については、1996年7月CQ出版株式会社発行の『入門Verilog-HDL記述』などに記載されている。

【0095】case x及びcaseは、それに続く()内の変数(opcode、tmg)に応じて、後続の値(16' b01010100_0001?????:、4' b0001: など)に一致した動作を実行する。begin～endで1つの動作が記述される。

【0096】(1)で、命令デコーダの入力(opcode)による条件を指示する。当該条件で、(2)～(12)でRTS/2に相当する命令コードの動作が記述される。更に、各命令コード毎に、ステップに相当する命令デコーダの入力(tmg)による条件を指示する。RTS/2の第1ステップの動作は、(4)～

(7)に記述される。(5)で、次のステップ(nexttmg)が第4ステップであることが指定される。nexttmgは1ステート後に、tmgとして命令デコーダに入力される。(6)で、バスコマンドがロングワードリードとされる。long_readはパラメータとして、所定のコードが割当てられるようにしておく。この後に、そのほかの制御信号が記述される。同様に、第4ステップの動作は、(8)～(11)に記述される。第5～第8ステップの動作は、これ以降(12)までに記述される。

【0097】同様に、命令デコーダの入力(opcode)による条件で、(13)以降でRTS/3に相当する命令コードの動作が記述される。第1ステップの動作記述において、(15)で、次のステップ(nexttmg)が第2ステップであることが指定される。また、第2ステップの動作は、(17)以降に記述される。更

に、第4～第8ステップの動作も記述される。

【0098】更に、その後、RTS/4に相当する命令コードの動作が記述され、第2の復帰命令(RTE/n)、複数レジスタの退避命令(STM)や、その他の命令コードも記述される。

【0099】図8に、第1の復帰命令(RTS/n)の実行タイミングを示す。

【0100】n=2の場合の例である。

【0101】特に制限はされないものの、内蔵ROM、RAMのリード/ライトを1ステートでリード/ライト可能とし、内蔵ROM上にプログラム、内蔵RAM上にスタックを配置した場合のタイミングを示す。

【0102】T0のφ#(φは反転論理を示す)で、CPUのアドレスバッファ(MAB)かアドレスIABに出力される。また、命令デコードから、命令フェッチ(if)を示す、バスコマンド(BCMD)が出力される。

【0103】T1のφで、IABの内容がPABに出力され、バスコマンドに基づき、リードサイクルが開始され、PDBにデータが出力される。φ#でPDBのリードデータが内部データバスIDBに得られ、これをT2のφでIRにラッチする。以上の動作は以前の命令の実行の制御によって行われる(プリフェッチ)。ここで、内蔵ROM、RAMは、PAB及びPDBに接続されていないが、PAB、PDB相当の動作をモジュール内で行い、本タイミング図には、このモジュール内の動作を示している。

【0104】直前の命令の実行が終了すると、最も早く命令の実行が開始される場合には、T2のφで命令コードがDECに入力されて、命令の内容が解読される。解読結果に従って、制御信号を出力して、各部の制御を行う。命令の一部(レジスタ指定フィールド:r)がレジスタセレクトに与えられる。

【0105】前記のステップ1として、T2のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。

【0106】T3のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T3のφ#から、スタックからリードしたデータがIDBに出力され、T4のφで、DBRに格納される。

【0107】ステップ4として、T4のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T3のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T4のφで、INCでインクリメント

(+4)された結果が、内部バスWBを経由してSPにライトされる。T4のφ#から、スタックからリードしたデータがIDBに出力され、T5のφで、DBRに格納される。

【0108】ステップ5として、T5のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T4のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T5のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T5のφ#から、スタックからリードしたデータがIDBに出力され、T6のφで、DBRに格納される。

【0109】ステップ6として、T6のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、PCに書込まれる。

【0110】ステップ7として、T6のφ#で、PCの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。T7のφで、INCでインクリメント(+2)された結果が、内部バスWBを経由してPCにライトされる。

【0111】ステップ8として、T7のφ#で、PCの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。T8のφで、INCでインクリメント(+2)された結果が、内部バスWBを経由してPCにライトされる。T8のφでリードした命令コードが、IRを経由して、DECに入力されて、次の命令が実行される。

【0112】図9に、第2の復帰命令(RTE/n)の動作フローを示す。

【0113】第1の復帰命令(RTS/n)に対して、ステップ6で、PCとともに、CCRへの書込みが行われるようにされる。そのほかの動作は同様にできる。

【0114】図10に、第2の復帰命令(RTE/n)の実行タイミングを示す。

【0115】n=4の場合の例である。

【0116】前記同様に、T2のφで命令コードがDECに入力されて、命令の内容が解読される。解読結果に従って、制御信号を出力して、各部の制御を行う。命令の一部(レジスタ指定フィールド:r)がレジスタセレクトに与えられる。

【0117】前記のステップ1として、T2のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。

【0118】T3のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライ

トされる。T3のφ#から、スタックからリードしたデータがIDBに出力され、T4のφで、DBRに格納される。

【0119】ステップ2として、T4のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T3のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T4のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T4のφ#から、スタックからリードしたデータがIDBに出力され、T5のφで、DBRに格納される。

【0120】ステップ3として、T5のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T4のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T5のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T5のφ#から、スタックからリードしたデータがIDBに出力され、T6のφで、DBRに格納される。

【0121】ステップ4として、T6のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T5のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T6のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T6のφ#から、スタックからリードしたデータがIDBに出力され、T7のφで、DBRに格納される。

【0122】ステップ5として、T7のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、指定(r)した汎用レジスタに書込まれる。レジスタ指定フィールドがデクリメントされる。前記同様に、T6のφ#で、SPの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。ロングワードのリードのバスコマンドが出力される。T7のφで、INCでインクリメント(+4)された結果が、内部バスWBを経由してSPにライトされる。T7のφ#から、スタックからリードしたデータがIDBに出力され、T6のφで、DBRに格

納される。

【0123】ステップ6として、T6のφ及びφ#で、DBRの内容が内部バスGB、ALU、内部バスWBを介して、CCRとPCに書込まれる。DBRのビット31~24がCCRに、ビット23~0がPCに書込まれる。

【0124】ステップ7として、T8のφ#で、PCの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。T9のφで、INCでインクリメント(+2)された結果が、内部バスWBを経由してPCにライトされる。

【0125】ステップ8として、T9のφ#で、PCの内容を内部バスGBに読み出して、ABとINCに入力する。ABからアドレスIABが出力される。T10のφで、INCでインクリメント(+2)された結果が、内部バスWBを経由してPCにライトされる。T10のφでリードした命令コードが、IRを経由して、DECに入力されて、次の命令が実行される。

【0126】図11に、複数汎用レジスタの退避命令(STM)の動作フローを示す。

【0127】ステップ0は、命令コードの第1ワードで制御される。制御信号Aで、プログラムカウンタ(PC)の内容のアドレスバス(IAB)への出力と、メモリワードリードが指示され、PCの内容のインクリメント(+2)が指示される。制御信号Bで、命令コードの第2ワードの命令デコーダへの入力と、第1ワードで指定される汎用レジスタ本数の情報(MOD)が指示される。

【0128】ステップ1で、制御信号Aで、スタックポインタ(SP)の内容のデクリメント(-4)が指示される。ステップ2~5で、制御信号Aで、スタックポインタ(SP)の内容のアドレスバス(IAB)への出力と、ロングワードのライトが指示される。制御信号Bで、汎用レジスタの内容のDBWへの格納が指示される。制御信号Cで、DBWの内容のIDBへの出力が指示される。レジスタデコーダRDECに対してレジスタフィールドのインクリメント(-1)が指示される。ステップ2~4は、SPの内容のデクリメント(-4)が指示される。ステップ2~5は、指定した汎用レジスタの本数(n)に従って、所定回数実行される。

【0129】ステップ6で、制御信号Aで、プログラムカウンタ(PC)の内容のアドレスバス(IAB)への出力と、メモリワードリードが指示され、PCの内容のインクリメント(+2)が指示される。制御信号Bで、前のステップでリードした命令(IR)の命令デコーダへの入力が指示される。次の命令の実行が開始される。

【0130】ステップ2~5において、汎用レジスタの内容を、DBWに転送後、当該汎用レジスタの内容を0にクリアするようにするとよい。汎用レジスタを退避した後、0にクリアし、その内容を確定することで、その

とサブルーチン分岐命令にコンパイルされる。

【0149】アセンブラは、アセンブリ言語ソースプログラムを入力し、オブジェクトモジュールを出力する。

【0150】リンカージェディタは、上記Cコンパイラやアセンブラの生成した、複数のオブジェクトモジュールを入力して、各モジュールの外部参照や相対アドレスなどの解決を行い、1つのプログラムに結合して、ロードモジュールを出力する。

【0151】ロードモジュールは、シミュレータ/デバグガに入力して、パーソナルコンピュータなどのシステム開発装置上で、CPUの動作をシミュレーションし、実行結果を表示し、プログラムの解析や評価を行うことができる。また、エミュレータに入力して、実際の応用システム上などで動作する、いわゆるインサーキットエミュレーションを行ない、マイクロコンピュータ全体としての、実動作の解析や評価を行うことができる。

【0152】更に、ロードモジュールをPROMライタなどのROM書き込み装置に入力して、マイクロコンピュータの内蔵ROMがフラッシュメモリなどの場合や、外部のフラッシュメモリなどに、作成したプログラムをロードすることができる。或いはマイクロコンピュータの製造工程で、内蔵ROMに書き込むことも可能である。必要に応じて、オブジェクトコンバータなどによって、所望のフォーマットに変換する。

【0153】このほかに、ライブラリアンとして、汎用的なサブルーチンなどを提供することもできる。

【0154】CPU定義情報ファイルは、コンパイル対象のCPUを指定し、本発明の復帰命令を持たない(既存の)CPUと、本発明のCPUとを共通の開発環境を利用可能にする。

【0155】例えば、CPU定義情報ファイルに、
SET CPU= CPU1;

と記述することによって、本発明のCPUを選択するようにする。

【0156】コンパイル時に、かかるCPU定義情報ファイルを参照し、本発明の復帰命令を持たないCPUが選択されていれば、前記関数は、

```
_seq0:
    STM      ER3-4, @-SP
    MOV. L   #0, ER3
    MOV. L   #0, ER4
    ...
    LDM      @SP+, ER3-4
    RTS
```

となる。MOV. Lは、当該関数の処理内容に従って、変更になる場合もある。

【0157】Cコンパイラ自体には、C言語によるプログラムを、CPUの命令に変換する機能のほか、C++言語によるプログラムのコンパイルや、モジュール間最適化などといった、CPUの命令セットとは直接関係の

ない機能の向上が図られているが、CPU毎の個別のコンパイラでは、これらの機能向上を全ての個別のコンパイラに適用しなければならない。本発明のように、共通のCコンパイラとしておけば、前記、CPUの命令セットとは直接関係のない機能向上を図ることが容易になり、また、開発効率などを向上することができる。

【0158】上記実施例によれば、以下の作用効果を得るものである。

【0159】単一の命令で、複数の汎用レジスタと復帰動作と、サブルーチンまたは例外処理ルーチンからの復帰動作を行うことによって、相対的に、命令コード長を短縮(例えば、2ワード短縮)し、命令のリード回数を低減して、高速化することができる。

【0160】例えば、C言語などで多用される関数はサブルーチンとされるから、この数に対応する数のサブルーチンからの復帰命令が使用されることになる。関数毎に必要なデータが汎用レジスタに割当てられるから、それ以前の汎用レジスタの内容の退避/復帰を伴う。従って、かかる復帰命令による命令コード長の短縮効果も大きくなる。

【0161】また、関数毎に、退避/復帰する汎用レジスタ本数も異なるから、レジスタの本数の異なる命令を複数命令サポートすることによって、或いは、汎用レジスタの復帰を伴わない復帰命令もサポートして、プログラムの作成を容易にし、使い勝手を向上することができる。

【0162】CPUの有する汎用レジスタ本数が多い場合にも、復帰する汎用レジスタを指定可能にしているから、汎用レジスタの使用方法的自由度を損なうことが少ない。また、不所望の汎用レジスタの退避/復帰を行ったりすることがない。

【0163】複数の汎用レジスタを指定する固定の組合わせにすることによって、命令コード長を短縮でき、更に、各命令の実行ステート数を固定にすることにより、内部の条件分岐を行うことを少なくし(命令デコードの入力opcodeに応じて、デコードを記述すればよい)、内部論理を簡潔にし、論理規模を縮小できる。マイクロプログラムによらず、ワイアードロジックなどでも容易に実現できる。マイクロプログラムによらず、ワイアードロジックなどとする事により、論理回路の高速化に寄与することができる。レジスタデコードにインクリメントを持つことによって、論理的規模の増加を最小限にして、連続した任意の組合せを指定可能にできる。

【0164】同様の指定を可能にした、汎用レジスタの退避命令(STM)を持ち、これをサブルーチン乃至例外処理ルーチンの先頭で実行し、最後に、本発明の復帰命令を実行することによって、より一層の使い勝手を向上できる。汎用レジスタの退避命令実行時に、退避後に汎用レジスタをクリアすることにより、更に、命令コー

ド長を短縮し、実行ステート数を短縮できる。

【0165】また、データのリード/ライトを連続して行うことによって、メモリに対するバースト動作などやバス幅を拡張する場合にも、これらを有効に利用しやすい。

【0166】ソースプログラムレベルまたはオブジェクトプログラムレベルで、既存のCPUの命令セットを包含した上で、上記命令を追加することによって、ソフトウェア資産を有効に利用することができ、使用者のソフトウェア開発効率を向上することができる。ソースプログラムレベルまたはオブジェクトプログラムレベルで互換性を保つことによる利点と前記転送命令を追加することの利点の双方を享受することができる。

【0167】以上本発明者等によってなされた発明を実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。

【0168】命令コードは一例であり、種々変更が可能である。既存の命令セットに追加するほか、新規の命令セットを開発する場合に適用することが可能であることは言うまでもない。

【0169】例外処理におけるスタックの構造は、CCRとPCを組合せて32ビットとしたが、これらは、16ビットと32ビットに分割したり、そのほかのコントロールレジスタを追加したりすることもできる。PCも32ビットとしてもよい。

【0170】汎用レジスタの構成や本数など、或いは復帰命令で復帰する汎用レジスタの本数などは任意に変更できる。復帰する汎用レジスタの本数は1本でもよい。スタックに格納される順序なども任意に変更できる。汎用レジスタを復帰する場合、PC（及びCCR）と連続してリードする方が都合がよい。

【0171】インクリメント（RINC）も、インクリメント/デクリメントを行うほか、命令デコードの出力との加算/減算を行うようにしてもよい。本発明の復帰命令の場合、命令コードのレジスタ指定フィールドに対して、1/2/3を順次減算するようにしてもよい。命令コード中に含まれるレジスタ指定フィールドも、最初に指定する汎用レジスタに対応している必要は必ずしもない。前記加算/減算と組合せて、所望の汎用レジスタを指定できればよい。

【0172】復帰する汎用レジスタとして、スタックポインタであるER7が指定された場合は、ER7への書込みを禁止するとよい。スタックポインタの退避/復帰は意味がなく、所定のポイント動作を行えばよいからである。

【0173】汎用レジスタの退避命令実行時に、退避後に汎用レジスタをクリアする命令と、保持する命令の両方を持ってもよい。当該命令コードの第1ワードの所定のビットでこれを指定して、制御信号（MOD）を発生して、第2ワードの動作を変更するようにすればよい。

【0174】汎用レジスタのクリアの内容は0に限定されず、他の値にしてもよい。クリアする内容を指定する手段を別に設けてもよい。

【0175】命令の基本単位も16ビットに限定されない。8ビット単位、32ビット単位でも、本発明の復帰命令は適用可能である。

【0176】簡単のために、バス幅を32ビットとし、命令リードを16ビット単位としたが、命令リードを32ビット単位にして高速化することができる。このような例は、特願平11-167812に記載されている。また、バス幅を16ビットとして、ロングワードのリード/ライトを、ワード単位の2回のリード/ライトで実行するようにしてもよい。このような例は、特願平11-320518に記載されている。

【0177】マイクロコンピュータのその他の機能ブロックについても何等制約されない。

【0178】以上の説明では主として本発明者によってなされた発明をその背景となった利用分野であるマイクロコンピュータに適用した場合について説明したが、それに限定されるものではなく、その他のデータ処理装置にも適用可能であり、本発明は少なくとも、複数のレジスタ手段を有するデータ処理装置に適用することができる。

【0179】

【発明の効果】本題において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【0180】サブルーチンから、または例外処理からの復帰命令において、当該命令コードに複数の汎用レジスタを指定する情報を含み、当該命令コードの実行時に、プログラムカウンタの復帰に加えて、前記指定した汎用レジスタの復帰を行うようにすることにより、使い勝手を向上し、プログラム容量と実行ステート数を短縮できる。

【0181】更に、指定する汎用レジスタの組合せを連続したものにより、命令コード長を短縮でき、更に、各命令の実行ステート数を固定にすることにより、内部の条件分岐を行うことを少なくし、内部論理を簡潔にし、論理規模を縮小できる。16ビット単位の変長の命令コード体系を採用する場合にも、命令プリフェッチや命令デコードを簡易にしつつ、命令コードを有効に利用して、全体的なプログラム効率を向上できる。

【0182】同様の指定を可能にした、汎用レジスタの退避命令を持ち、これをサブルーチン乃至例外処理ルーチンの先頭で実行し、最後に、本発明の復帰命令を実行することによって、より一層の使い勝手を向上できる。

【0183】汎用レジスタの復帰と、プログラムカウンタなどの復帰を連続的に行うことによって、内部の動作を容易にし、高速化できる。

【0184】レジスタの本数の異なる復帰命令を複数サ

ポートすることによって、また、連続した汎用レジスタの指定を可能にして、更に、プログラムの作成を容易にし、使い勝手を向上することができる。

【図面の簡単な説明】

【図1】本発明の適用されたマイクロコンピュータのブロック図。

【図2】CPUに内蔵されている汎用レジスタ及び制御レジスタの構成例の図。

【図3】CPUの構成例の図。

【図4】レジスタセクタRDECの一部の詳細ブロック図。

【図5】複数汎用レジスタの復帰を含む第1の復帰命令(RTS/n)、第2の復帰命令(RTE/n)の命令フォーマット図。

【図6】複数汎用レジスタの復帰を含む第1の復帰命令

(RTS/n)、第2の復帰命令(RTE/n)のスタックの構成例の図。

【図7】第1の復帰命令(RTS/n)の動作フローの図。

【図8】第1の復帰命令(RTS/n)の実行タイミングの図。

【図9】第2の復帰命令(RTE/n)の動作フローの図。

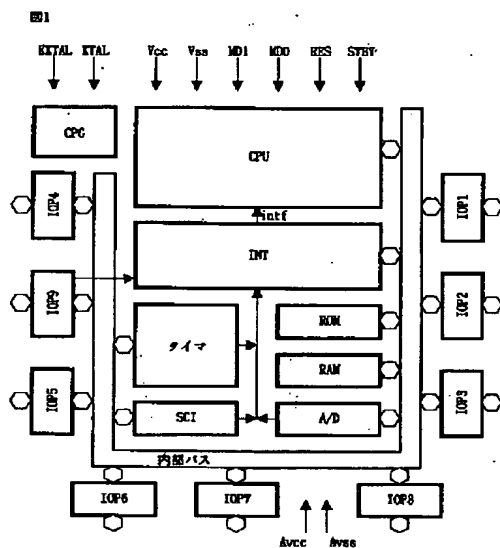
【図10】第2の復帰命令(RTE/n)の実行タイミングの図。

【図11】複数汎用レジスタの退避命令(STM)の動作フローの図。

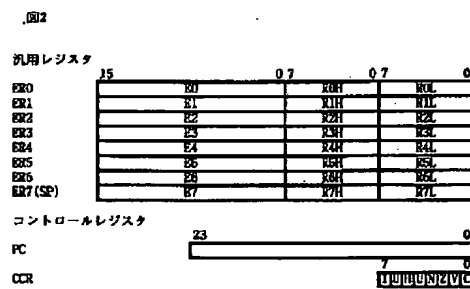
【図12】複数汎用レジスタの退避命令(STM)の実行タイミングの図。

【図13】本発明のCPUの開発環境の概略図。

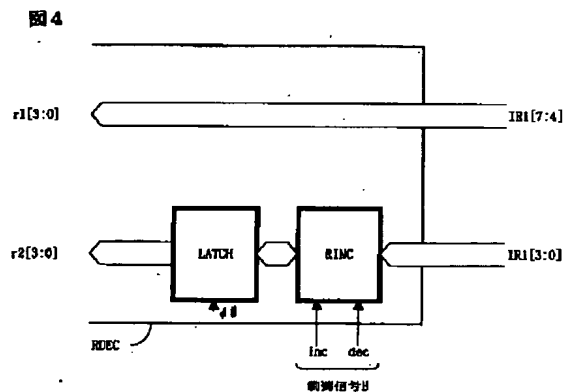
【図1】



【図2】



【図4】



【図3】

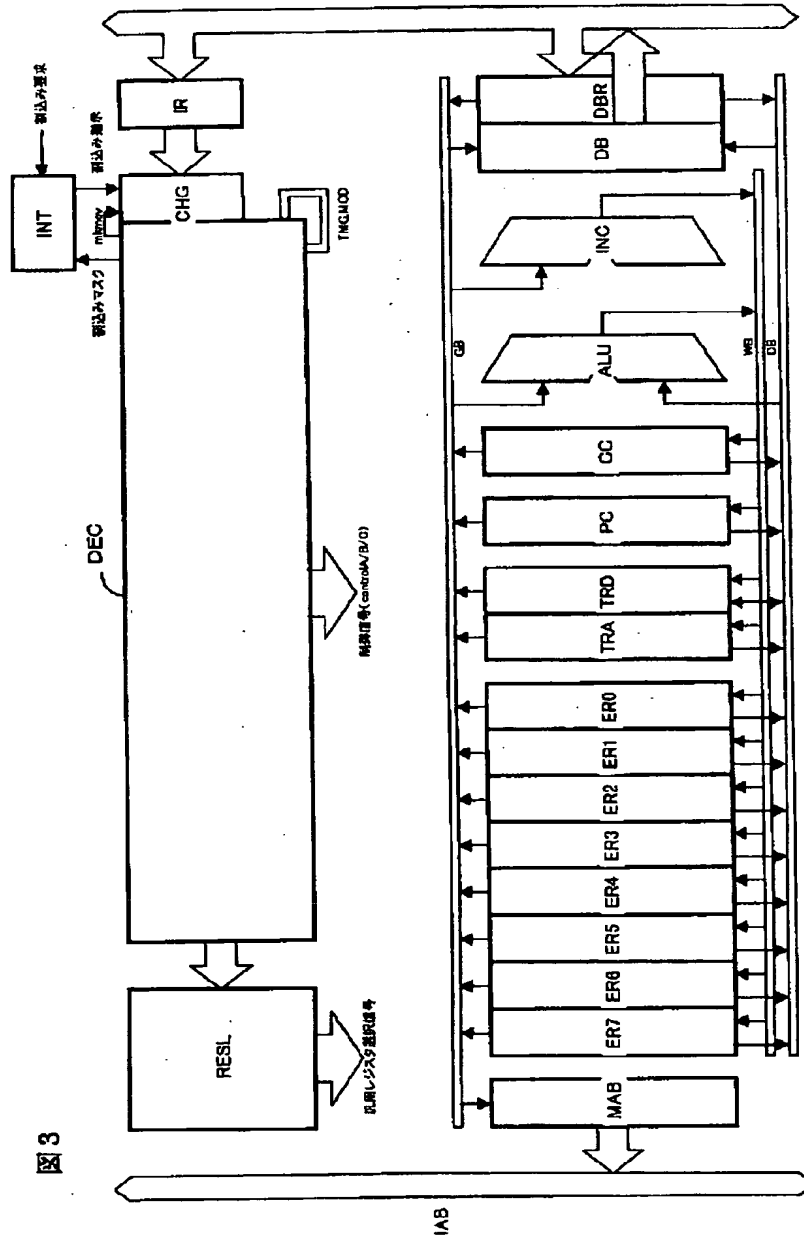
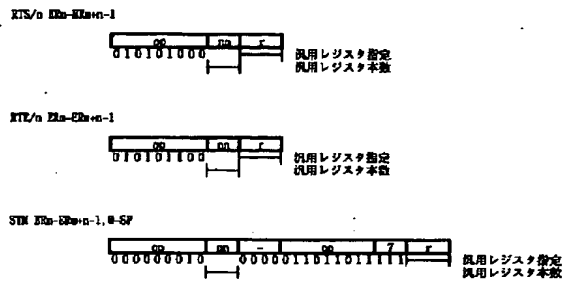


図 3

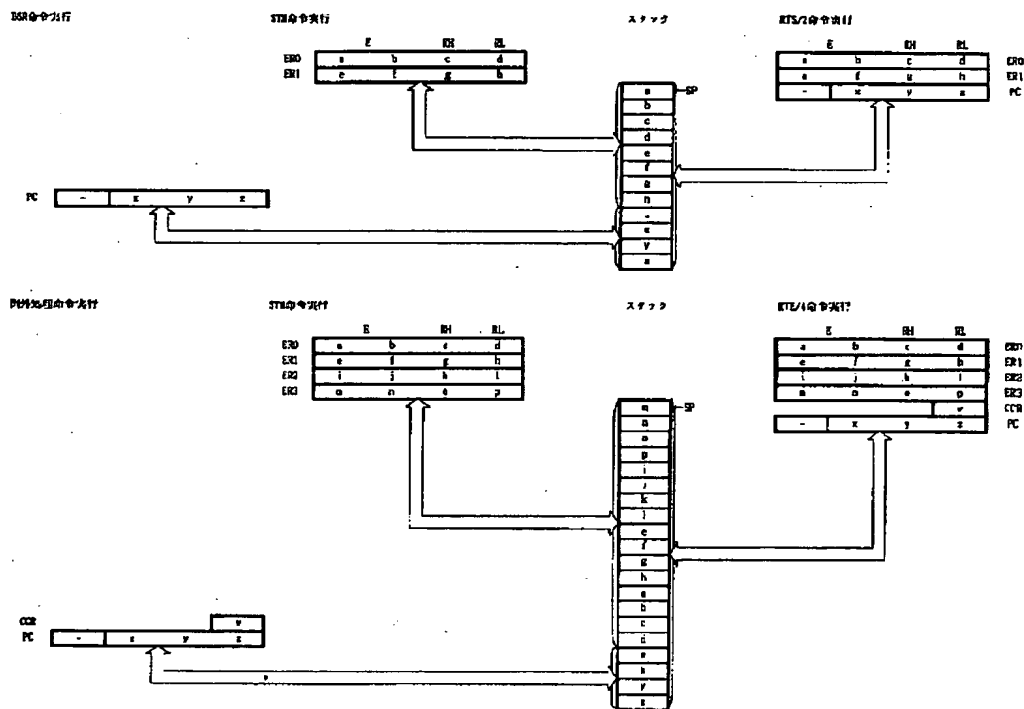
【图5】

圖 5



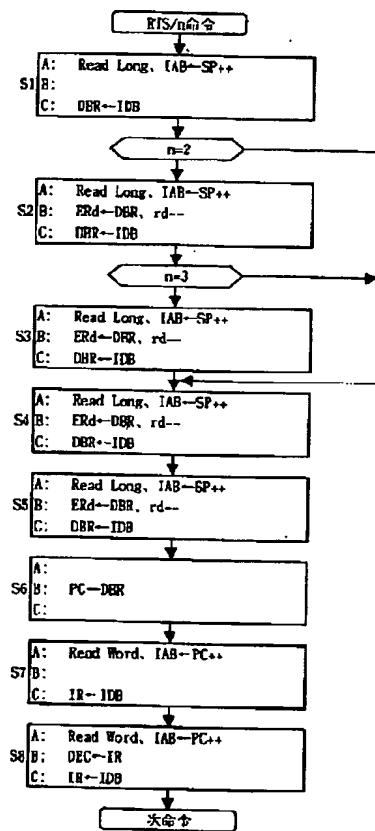
【図6】

图 6



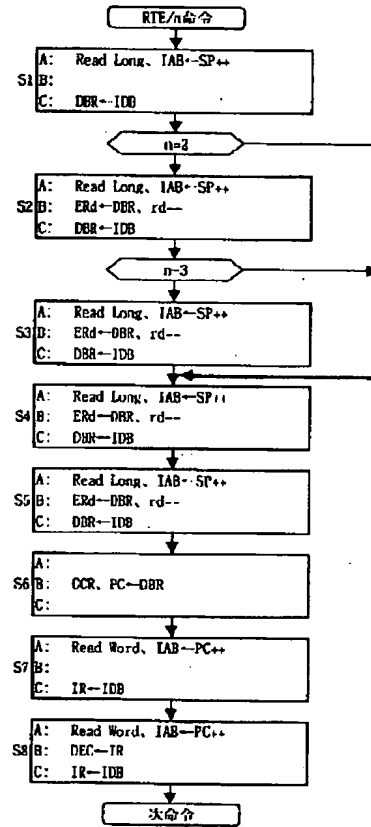
【図7】

図 7



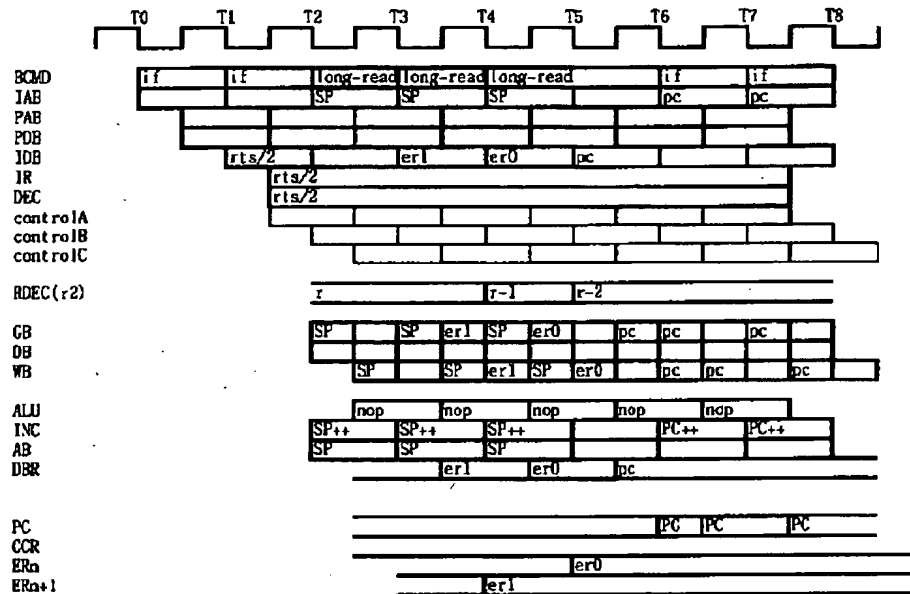
【図9】

図 9



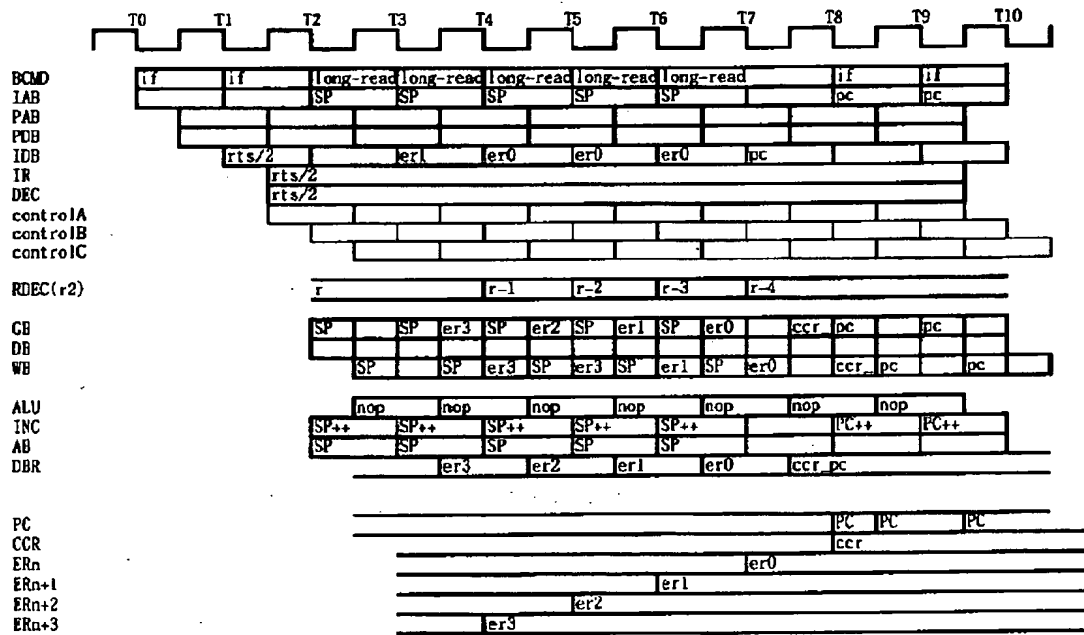
【図8】

図8 RTS/2



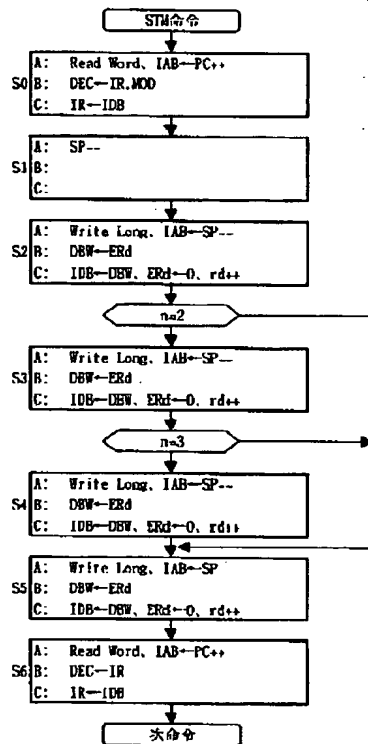
【図10】

図10 RTE/4



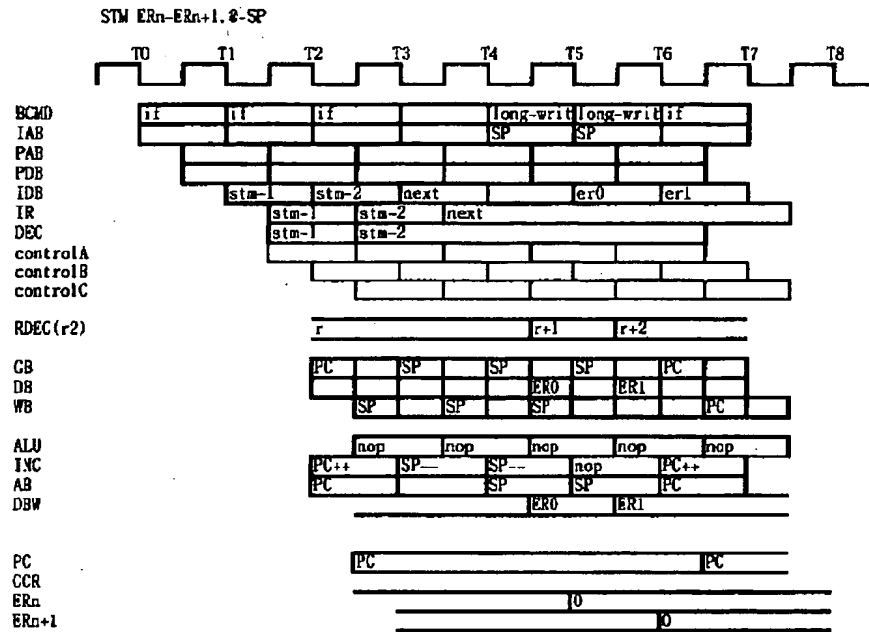
【図11】

図 11



【図12】

図12



【図13】

図13

